

Un sistema semplificato di gestione di un magazzino di prodotti alimentari può essere realizzato utilizzando le strutture seguenti:

```
struct prodotto
{
    char nome [LUNG]; int codice; int scad;
};
struct magazzino
{
    prodotto prod[NUM]; int numProd;
};
```

con **NUM** il numero massimo di prodotti contenuti nel magazzino e **numProd** il numero di prodotti effettivamente presenti nel magazzino. Ogni prodotto è individuato dal suo nome (al massimo di **LUNG** caratteri), da un codice e dalla scadenza che è espressa come numero di giorni a partire dall'inizio dell'anno. Quando è creata un'istanza di tipo **magazzino**, il valore di **numProd** è inizializzato a zero.

Scrivere il corpo delle seguenti funzioni C++.

1. **bool build(const char nome[], magazzino* m)** che utilizza il contenuto del file il cui nome è passato come primo argomento per inizializzare l'oggetto di tipo **magazzino** puntato da **m**; il file contiene un numero intero che rappresenta il numero di righe contenute nel file: ogni riga è costituita da un numero intero *n*, una stringa ed *n* coppie di numeri interi. La stringa indica il nome di un prodotto, la coppia indica il codice del prodotto e la scadenza del prodotto. Un esempio di riga è il seguente:

```
2      AAA      1 35      2 25
```

dove **2** indica che ci sono due prodotti il cui nome è **AAA**, il primo prodotto ha codice **1** e scadenza **35**, il secondo ha codice **2** e scadenza **25**. Tutti i prodotti che hanno lo stesso nome sono presenti in un'unica riga (ovvero prodotti con lo stesso nome non possono essere su due righe diverse); conseguentemente prodotti con lo stesso nome sono posti consecutivamente nel campo **prod** della struttura **magazzino**.

La funzione restituisce **false** se il numero di prodotti supera la dimensione massima del vettore **prod** e se si verificano errori nella lettura del file; altrimenti restituisce **true**;

2. **void ord(prodotto vettore[], int inf, int sup)** che ordina per valori crescenti del membro **scad** gli elementi del vettore (passato come argomento alla funzione) con indici superiori o uguali ad **inf** e inferiori o uguali a **sup**;
3. **void ordina(magazzino* m)** che ordina i prodotti con lo stesso nome per valori crescenti del membro **scad** (si ricorda che prodotti con lo stesso nome si trovano in posizioni consecutive nell'array **prod**); si utilizzi la funzione **ord()** realizzata nel punto precedente;
4. **bool elimina(magazzino* m, const char nome [])** che elimina il prodotto che scade prima tra quelli il cui nome coincide con il nome passato come argomento formale (si supponga che questa funzione venga chiamata dopo aver chiamato la funzione **ordina()**); dopo aver eliminato l'elemento si compatti il campo **prod** della struttura **magazzino** puntata da **m**.
5. **int piuScad(magazzino* m)** che restituisce l'indice nell'array **prod** della struttura **magazzino** puntata da **m** del prodotto che scade per primo; si supponga che il magazzino contenga almeno un prodotto.

NOTE SULLO SVOLGIMENTO DELLA PROVA:

- PRIMA DI INIZIARE LO SVOLGIMENTO DELL'ELABORATO, POSIZIONARSI nel directory **c:\esame\esainf** ed eseguire il programma **esame.exe**, rispondendo alle domande;
- Per svolgere l'elaborato, lanciare il DevC++, aprire il progetto *esaInf.dev* presente nel directory *c:\esame\esaInf* e scrivere le funzioni richieste nel file *compito.cpp*, già presente nel progetto;
- per una corretta stampa dell'elaborato bisogna mantenere il codice entro i margini imposti dall'ambiente Dev-C++ (linea verticale presente alla destra della pagina);
- per la produzione dei risultati, aprire una finestra DOS, digitare **cd c:\esame\esaInf** e premere il tasto INVIO; quindi digitare **esaInf.exe > output.txt** e premere più volte il tasto INVIO finché non compare di nuovo il prompt del DOS.